

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра теорії оптимальних процесів**

**Затверджено**

На засіданні кафедри теорії оптимальних процесів факультету прикладної математики та інформатики Львівського національного університету імені Івана Франка

(протокол № 1 від 28.08.2024 р.)

Завдувач кафедри Шахно С.М.



**Силабус з навчальної дисципліни**  
**“Сучасні технології програмування”,**  
**що викладається в межах ОПП «Системний аналіз» другого**  
**(магістерського) рівня вищої освіти для здобувачів з**  
**спеціальності 124 – системний аналіз**

Львів 2024 р.

<b>Назва дисципліни</b>	Сучасні технології програмування
<b>Адреса викладання дисципліни</b>	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики Кафедра теорії оптимальних процесів
<b>Галузь знань, шифр та назва спеціальності</b>	12 – інформаційні технології 124 – системний аналіз
<b>Викладачі дисципліни</b>	Недашковський Микола Олександрович, професор кафедри теорії оптимальних процесів, Голуб Богдан Михайлович, доцент кафедри теорії оптимальних процесів, Шунькін Юрій Валерійович асистент кафедри теорії оптимальних процесів.
<b>Контактна інформація викладачів</b>	<a href="mailto:mykola.nedashkovskyy@lnu.edu.ua">mykola.nedashkovskyy@lnu.edu.ua</a> ; <a href="https://ami.lnu.edu.ua/employee/nedashkovskyy-mykola-oleksandrovych">https://ami.lnu.edu.ua/employee/nedashkovskyy-mykola-oleksandrovych</a> ; <a href="mailto:bohdan.holub@lnu.edu.ua">bohdan.holub@lnu.edu.ua</a> ; <a href="https://ami.lnu.edu.ua/employee/holub">https://ami.lnu.edu.ua/employee/holub</a> <a href="mailto:yuriy.shunkina@lnu.edu.ua">yuriy.shunkina@lnu.edu.ua</a> Головний корпус ЛНУ ім. І. Франка, каб. 269. м. Львів, вул. Університетська, 1
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/практичних занять (за попередньою домовленістю).
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/suchasni-tekhnohii-prohramuvannia">https://ami.lnu.edu.ua/course/suchasni-tekhnohii-prohramuvannia</a>
<b>Інформація про дисципліну</b>	Курс розроблено таким чином, щоб надати учасникам базові знання сучасних технологій програмування, перелік яких формується на основі рейтингу використання алгоритмічних мов
<b>Коротка анотація дисципліни</b>	Дисципліна “Сучасні технології програмування” є нормативною дисципліною зі спеціальності 124 – системний аналіз для освітньої програми Системний аналіз, яка викладається у 1-му семестрі в обсязі 3-х кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Мета та цілі дисципліни</b>	Метою вивчення нормативної дисципліни “Сучасні технології програмування” є надання студентам базових знань та практичних навичок використання сучасних технологій програмування.
<b>Література для вивчення дисципліни</b>	1. Основи JavaScript / <a href="https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics">https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics</a> 2. Java Підручник / <a href="https://w3schoolsua.github.io/java/index.html#gsc.tab=0">https://w3schoolsua.github.io/java/index.html#gsc.tab=0</a> 3. Python Підручник / <a href="https://w3schoolsua.github.io/python/index.html#gsc.tab=0">https://w3schoolsua.github.io/python/index.html#gsc.tab=0</a> 4. PHP Manual / <a href="https://www.php.net/manual/en/index.php">https://www.php.net/manual/en/index.php</a> 5. Б.М.Голуб. С#. Концепція та синтаксис. Навчальний посібник. – Львів: ВНТЛ, 2016. – 144с. 6. CSS first steps / <a href="https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps">https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps</a>

	<p>7. Ruby – Вікіпідручник / <a href="https://uk.wikibooks.org/wiki/Ruby">https://uk.wikibooks.org/wiki/Ruby</a></p> <p>8. Swift. Посібник з мови / <a href="https://book.swift.org.ua/book/1_language_guide">https://book.swift.org.ua/book/1_language_guide</a></p> <p>9. A Tour of Scala / <a href="https://docs.scala-lang.org/tutorials/">https://docs.scala-lang.org/tutorials/</a></p> <p>10. Командний інтерпретатор Shell / <a href="http://ni.biz.ua/17/17_11/17_119813_komandniy-interpretator-SHELL.html">http://ni.biz.ua/17/17_11/17_119813_komandniy-interpretator-SHELL.html</a></p> <p>11. K.Seguin. The Little Go Book / <a href="https://www.openmymind.net/assets/go/go.pdf">https://www.openmymind.net/assets/go/go.pdf</a></p> <p>12. Освоюємо R / <a href="https://uk.wikibooks.org/wiki/%D0%9E%D1%81%D0%B2%D0%BE%D1%8E%D1%94%D0%BC%D0%BE_R">https://uk.wikibooks.org/wiki/%D0%9E%D1%81%D0%B2%D0%BE%D1%8E%D1%94%D0%BC%D0%BE_R</a></p> <p>13. TypeScript. Basic Types / <a href="https://www.typescriptlang.org/docs/handbook/basic-types.html">https://www.typescriptlang.org/docs/handbook/basic-types.html</a></p> <p>14. Kotlin docs / <a href="https://kotlinlang.org/docs/home.html">https://kotlinlang.org/docs/home.html</a></p> <p>15. Introduction to Dart / <a href="https://dart.dev/language">https://dart.dev/language</a></p> <p>16. TypeScript. Підручник / <a href="https://w3schoolsua.github.io/typescript/index.html#gsc.tab=0">https://w3schoolsua.github.io/typescript/index.html#gsc.tab=0</a></p> <p>17. Типи баз даних: особливості, відмінності та приклади / <a href="https://dou.ua/lenta/articles/types-of-databases/">https://dou.ua/lenta/articles/types-of-databases/</a></p>
<b>Обсяг курсу</b>	Загальний обсяг: 80 годин. Аудиторних занять: 48 год., з них 16 год. лекцій та 32 години лабораторних робіт. Самостійної роботи: 32 год.
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде :</p> <p><b>Знати</b> основи технологій: JavaScript, Java, Python, PHP, C#, CSS, Ruby, Swift, Scala, Shell, Go, Dart, TypeScript, Perl, Kotlin</p> <p><b>Вміти:</b> застосовувати набуті знання для розробки програмного забезпечення.</p> <p><b>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</b></p> <p><b>Загальні компетентності (ЗК) і спеціальні (фахові, предметні) компетентності (СК):</b></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК3. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК6. Здатність застосовувати теорію і методи Data Science для здійснення інтелектуального аналізу даних з метою виявлення нових властивостей та генерації нових знань про складні системи.</p> <p>ЗК7. Здатність проявляти лідерські навички, мотивувати людей, працювати в команді.</p> <p>ЗК8. Здатність свідомо та соціально-відповідально діяти на основі етичних міркувань і принципів академічної доброчесності.</p> <p>СК1. Здатність інтегрувати знання та здійснювати системні дослідження, застосовувати методи математичного та інформаційного моделювання складних систем та процесів різної природи.</p> <p>СК2. Здатність проєктувати архітектуру інформаційних систем.</p> <p>СК7. Здатність управляти робочими процесами у сфері інформаційних технологій, які є складними, непередбачуваними та потребують нових стратегічних підходів.</p> <p>СК8. Здатність розробляти і реалізовувати наукові та прикладні проєкти в галузі інформаційних технологій та дотичні до неї міждисциплінарні проєкти.</p>

<b>Ключові слова</b>	JavaScript, Java, Python, PHP, C#, CSS, Ruby, Swift, Scala, Shell, Go, Dart, TypeScript, Perl, Kotlin					
<b>Формат курсу</b>	Очний, дистанційний Проведення лекцій, лабораторних робіт і консультацій.					
<b>Теми</b>	Подано нижче у таблиці Схема курсу “Сучасні технології програмування”					
<b>Підсумковий контроль, форма</b>	Екзамен у кінці семестру					
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з - Програмування;					
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Презентації, лекції Індивідуальні завдання Групові проекти, менторство					
<b>Необхідне обладнання</b>	Комп’ютер та Internet-доступ до хмарного середовища (Amazon/Windows Azure/Google (App Engine)).					
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	Оцінювання проводиться за 100-бальною шкалою.					
	<b>Оцінка за шкалою ECTS</b>		<b>Оцінка в балах</b>	<b>Оцінка за національною шкалою</b>		
				<b>Екзамен, диференційований залік</b>	<b>залік</b>	
	A	Відмінно	100 - 90	Відмінно	5	зараховано
	B	Дуже добре	81- 89	Добре	4	
	C	Добре	71 -80			
	D	Задовільно	61 - 70	Задовільно	3	
	E	Достатньо	51- 60			
FX (F)	Незадовільно	0 - 50	Незадовільно	2	не зараховано	
<p>Впродовж семестру студент може отримати 100 балів. З них:</p> <ul style="list-style-type: none"> <li>- <b>за роботу на лабораторних заняттях:</b> максимальна кількість – 70 балів (7 програм (індивідуальні завдання) на вільний вибір (див. Теми курсу) по 10 балів); для кожного завдання встановлено терміни здачі. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (кожне лабораторне заняття на 10% балів менше).</li> <li>- залікова <b>контрольна робота:</b> максимальна кількість – 30 балів (3 теоретичних завдання по 10 б.).</li> </ul> <p>Підсумкова максимальна кількість балів 100.</p>						

Поточне тестування та самостійна робота			Контр. робота	Залік	Сума
Індивідуальне завдання №1	...	Індивідуальне завдання №7	30		100
10	...	10			

**Критерії оцінювання індивідуальних завдань:**

Кожне лабораторне завдання має містити демонстрацію 10 функціональностей по вибраній темі (напр. оператори, функції, потоки, робота з файлами, серіалізація тощо)

**10 балів** – студент повністю виконав умови завдання, відповідає на всі запитання, пов'язані з тематикою завдання;

**8-9 балів** – студент повністю виконав умови завдання, на деякі запитання, пов'язані з тематикою завдання, відповідає з незначними неточностями;

**6-7 балів** – студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач;

**4-5 бали** – студент виконав завдання частково (не менше 50%), проте демонструє уміння доповнення функціональності програми;

**2-3 бали** – студент виконав завдання частково (30-50%), проте демонструє уміння доповнення функціональності програми;

**1 бал** – студент виконав завдання частково (менше 30%), або не демонструє уміння доповнення функціональності програми;

**0 балів** – студент не виконав завдання.

**Академічна доброчесність:**

Відвідування лекцій та лабораторних завдань обов'язкове.

Жодні форми порушення академічної доброчесності не толеруються.

**Питання до заліку чи екзамену.**

- JavaScript
- Java
- Python
- PHP
- C#
- CSS
- Ruby
- Swift
- Scala
- Shell
- Go
- Dart
- TypeScript
- PowerShell
- Типи баз даних
- Kotlin

**Опитування**

Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

## Схема курсу “Сучасні технології програмування”

Тиждень	Тема, план, короткі тези	Форма діяльності (заняття)	Література, ресурси в інтернеті	Завдання, год.	Термін виконання
1	<b>Тема 1. Алгоритмічна мова JavaScript:</b> <i>Структура мови. Ядро (ECMAScript). Об'єктна модель браузера (BOM). Об'єктна модель документа (DOM). Обробка подій. Відокремлення від розмітки. Букмарклетки. Користувацькі скрипти в браузері. Середовища розробки</i>	лекція (1 год)	[1]		
	<b>Тема 2. Алгоритмічна мова Java:</b> <i>Синтаксис та ООП. Винятки. Графічний інтерфейс користувача. Аплети. Колекції. Менеджери розташування. Графічні компоненти Swing. Потоки. Паралелізм. Серіалізація. Середовища розробки</i>	лекція (1 год)	[2]		
	<b>Тема 1. Приклад JavaScript-програми</b>	лабораторне (2 год)		Розробка JavaScript-програми (2 год.)	Під час заняття
2	<b>Тема 2. Приклад Java-програми (Колекції)</b>	лабораторне (2 год)		Розробка Java-програми (2 год.)	Під час заняття
3	<b>Тема 3. Алгоритмічна мова Python:</b> <i>Синтаксис та ООП. Інтерпретатор Пайтона. Обробка помилок. Виконання скриптів. Кодування джерела. Стартові файли для діалогового режиму. Середовища розробки</i>	лекція (1 год)	[3]		
	<b>Тема 4. Алгоритмічна мова C#:</b> <i>Синтаксис C#. Простори імен. Модифікатори доступу. Структури і класи. Інтерфейси.</i>	лекція (1 год)	[5]		

	<i>Делегати. Події. Директиви препроцесору. Серіалізація. Середовища розробки</i>				
	<b>Тема 2.</b> Приклад Java-програми (Потоки, Серіалізація)	лабораторне (2 год)		Розробка Java-програми (2 год.)	Під час заняття
<b>4</b>	<b>Тема 3.</b> Приклад Python-програми	лабораторне (2 год)		Розробка Python-програми (2 год.)	Під час заняття
<b>5</b>	<b>Тема 5. Алгоритмічна мова TypeScript:</b> <i>Текстові редактори TS та Visual Studio Code. Статична типізація. Інтерфейси. Класи. Примітиви. Модулі. Файли-декларації. Середовища розробки</i>	лекція (1 год)	[13,16]		
	<b>Тема 6. Алгоритмічна мова PHP:</b> <i>Синтаксис та типи даних. Змінні, вирази та оператори. Масиви. Функції. HTML-форми та cookies. Класи та об'єкти. Робота з файлами. Робота з БД. Середовища розробки</i>	лекція (1 год)	[4]		
	<b>Тема 4.</b> Приклад C#-програми	лабораторне (2 год)		Розробка C#-програми (2 год.)	Під час заняття
<b>6</b>	<b>Тема 5.</b> Приклад TypeScript-програми	лабораторне (2 год)		Розробка TypeScript-програми (2 год.)	Під час заняття
<b>7</b>	<b>Тема 7. Алгоритмічна мова Kotlin:</b> <i>Функції, змінні, класи, трейти, зовнішні функції. Система типів. Функції вищих порядків. Предметно-орієнтовані мови. Середовище розробки.</i>	лекція (1 год)	[14]		
	<b>Тема 8. Бази даних. SQL:</b> <i>Типи баз даних. Ієрархічні БД. Мережеві БД. Реляційні (PostgreSQL, MS SQL, MySQL, MariaDB ). Нереляційні БД. Документноорієнтовані</i>	лекція (1 год)	[17]		

	<i>(MongoDB). Графові (OrientDB)</i>				
	<b>Тема 6.</b> Приклад PHP-програми	лабораторне (2 год)		Розробка PHP-програми (2 год.)	Під час заняття
<b>8</b>	<b>Тема 7.</b> Приклад Kotlin-програми	лабораторне (2 год)		Розробка Kotlin-програми (2 год.)	Під час заняття
<b>9</b>	<b>Тема 9. Алгоритмічна мова Swift:</b> <i>Типи даних. Оператори. Кортежі. Функції. ООП у Swift. Колекції. Середовища розробки</i>	лекція (1 год)	[8]		
	<b>Тема 10. Алгоритмічна багатопотокова мова Go:</b> <i>Структура програми. Змінні, типи даних, оператори. Вказівники. Похідні типи. Пакети. Інтерфейси. Паралельне програмування. Мережеве програмування. Середовища розробки</i>	лекція (1 год)	[11]		
	<b>Тема 8.</b> Приклади запитів у базах даних PostgreSQL та MongoDB	лабораторне (2 год)		Підготовка баз даних (2 год.)	Під час заняття
<b>10</b>	<b>Тема 9.</b> Приклад Swift-програми	лабораторне (2 год)		Розробка Swift-програми (2 год.)	Під час заняття
<b>11</b>	<b>Тема 11. Алгоритмічна мова Ruby:</b> <i>Семантика. Контейнери. Процедурні об'єкти та ітератори. Класи, домішки, перевантаження операторів. Редактори Ruby. Стандартна бібліотека. Середовища розробки</i>	лекція (1 год)	[7]		
	<b>Тема 12. Алгоритмічна мова Dart:</b> <i>Крос-платформна мова Dart від Google. Основні конструкції, типи даних, алгоритмічні та функціональні можливості. Середовища розробки</i>	лекція (1 год)	[15]		

	<b>Тема 10.</b> Приклад Go-програми	лабораторне (2 год)		Розробка Go-програми (2 год.)	Під час заняття
<b>12</b>	<b>Тема 11.</b> Приклад Ruby-програми	лабораторне (2 год)		Розробка Ruby-програми (2 год.)	Під час заняття
<b>13</b>	<b>Тема 13. Алгоритмічна мова Scala:</b> Ієрархія типів Scala. Класи, трейти, кортежі. Функції вищого порядку. Вкладені методи. Множинні списки параметрів. Регулярні вирази. Оператори. Анотації. Середовища розробки	лекція (1 год)	[9]		
	<b>Тема 14. Мова опису зовнішнього вигляду сторінок CSS:</b> Синтаксис CSS. Селектори та псевдокласи. Приєднання до сторінки. CSS рівня 2. Медіа-типи. Модель візуального форматування. Візуальні ефекти.	лекція (1 год)	[6]		
	<b>Тема 12.</b> Приклад Dart-програми	лабораторне (2 год)		Розробка Dart-програми (2 год.)	Під час заняття
<b>14</b>	<b>Тема 13.</b> Приклад Scala-програми	лабораторне (2 год)		Розробка Scala-програми (2 год.)	Під час заняття
<b>15</b>	<b>Тема 15. Командний інтерпретатор Shell:</b> Структура команд. Групування команд. Генерація імен файлів. Shell-змінні. Параметри. Програмне середовище. Програмні структури. Функції в Shell.	лекція (2 год)	[10]		
	<b>Тема 14.</b> Приклади функціональності CSS	лабораторне (2 год)			Під час заняття
<b>16</b>	<b>Тема 15.</b> Приклади функціональності Shell	лабораторне (2 год)		Підготовка демонстраційного коду Shell (2 год.)	Під час заняття